

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPEAL BRIEF

Atty. Docket No.
VIGN1690-1

Applicant
Dean Moses

Application Number
10/091,513

Filed
03/07/2002

Title
Method and System for Sharing Different Web Components Between Different Web Sites in a Portal Framework

Group Art Unit
2152

Examiner
Dinh, Dung C.

Confirmation No.
8808

Mail Stop: Appeal
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Certification Under 37 C.F.R. §1.8

I hereby certify that this document is being deposited with the United States Postal Service as Express Mail to Addressee (Label No. EV616963215US) in an envelope addressed to: Mail Stop: Appeal Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22312-1450 on **February 14, 2005**.

Janice Pampell
Janice Pampell

Further to the Notice of Appeal filed October 13, 2004, Appellant presents this Appeal Brief. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

02/18/2005 HMARZ11 00000011 10091513

01 FC:1402

500.00 OP

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST	3
II. RELATED APPEALS AND INTERFERENCES	3
III. STATUS OF CLAIMS	3
IV. STATUS OF AMENDMENTS	3
V. SUMMARY OF THE INVENTION	4
VI. ISSUES	5
VII. GROUPING OF CLAIMS	5
VIII. ARGUMENT	5
1. Introduction	5
2. Prior Art	6
3. Improvement Over Prior Art	7
4. Examiner's / Appellant's Positions Regarding Obviousness	7
5. Rejections Under 35 U.S.C. §103	8
5.1 Examiner's Reasoning	8
5.2 Not All Limitations Disclosed	9
5.3 Flaws In The Examiner's Reasoning	11
5.4 Applying the System to Stefik to the Resources of Chang would Result in Needless Complexity	11
5.5 Examiner Has Failed To Show Suggestion Or Motivation To Combine References	13
5.6 Unacceptable Performance of System of Stefik Applied to Chang	14
5.7 Examiner Has Failed To Show Reasonable Expectation Of Success	14
5.8 Examiner Has Failed To Make A Prima Facie Case Of Obviousness Under 35 U.S.C. §103	15
6. Conclusion	15
IX. APPENDICES	17
Appendix A (Claims)	17

I. REAL PARTY IN INTEREST

The subject application is owned by Vignette, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at Corporation Trust Center, 1209 Orange Street Wilmington, DE 19801.

II. RELATED APPEALS AND INTERFERENCES

Appellants believe that there are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-39 of Appellant's Application serial no. 10/091,513 ("Application") were originally filed in the parent application. Claims 40-44 were added by Appellant's May 23, 2003 Response to Office Action. Claims 1, 14 and 40 were amended in Appellant's November 12, 2003 Response to Office Action. Claims 1-44 stand finally rejected. Claims 1-44, which are rejected under 35 U.S.C. §103, are the subject of this appeal. A copy of Claims 1-44 as on appeal is included in the Appendix hereto.

IV. STATUS OF AMENDMENTS

No amendments were filed subsequent to the final rejection. The first amendment to the application, filed on May 23, 2003, added Claims 40-44. The second amendment to the application, filed on November 12, 2003 amended Claims 1, 14 and 40. The Appendix hereto reflects the current state of the claims including the addition of Claims 40-44 and amendments to Claims 1, 14 and 40.

V. SUMMARY OF THE INVENTION

The invention comprises systems and methods for sharing objects between sites in a portal framework to allow more efficient administration of multiple sites that require access to common objects. (See Application, Summary of the Invention; Page 5, Lines 1, 8) These objects may be invocable software objects. As described, software objects encapsulate data and logical processes. (See Application, Page 10, Line 19-Page 11, Line 7) A software object may be invocable because the logical processes can run such that the software object represents actors in the system. A reference to a software object is a reference which points to the location of a software object in a database or file system and may identify the type of the invocable software object. Based on a reference to a software object, the software object can be located and an instance of the software object invoked, as, for example, by a programmatic user of a site.

One embodiment of the present invention is drawn to the duplication of a reference to an invocable software object from a first repository for a first site to a second repository. A second site that has access to the second repository has access to the reference to the invocable software object. Hence, the second site can invoke the software object using the reference to the software object, despite the fact that the second site may not have access to the first repository or the first site. (See Summary of the Invention, Page 17, Lines 2-35, Page 30, Line 14-Page 21, Line 15)

In certain embodiments, access to these objects can be controlled by requiring that the duplication of a reference to a software object occur according to a set of permissions. For example, if a first site wishes to share a reference to object A which is a reference in a first repository associated with the first site, the first site can only duplicate a reference to object A to a second repository associated with a second site if the first site has privileges to add references to the second repository associated with the second site. If the reference is duplicated to the second repository the second site can invoke an instance of object A even without access to the first repository. (See Page 28, Line 1-Page 31, Line 1)

VI. ISSUES

Whether claims 1-44 are unpatentable under 35 U.S.C. §103 over U.S. Patent No. 6,236,971 B1 ("Stefik") and Pub. No. US 2002/0078377 ("Chang").

VII. GROUPING OF CLAIMS

Claims 1-44 stand or fall together.

VIII. ARGUMENT

1 Introduction

The Examiner asserts that the present invention is obvious from the combination of the system of Stefik and the system of Chang. Thus, the Examiner asserts that the Stefik reference and the Chang reference teach all the limitations of the invention in question, and that a person of ordinary skill in the art of the invention would have combined the teachings of the Stefik's reference with the teachings of Chang to arrive at the invention.

The Appellant respectfully submits that not only do Stefik and Chang fail to teach all the limitations of the invention, but additionally, that the teachings of the Stefik and the Chang references are in distinct domains and that a person of ordinary skill in the art would not have combined references from these two domains as suggested by the Examiner. More particularly, the difference between the teachings of the Stefik reference and the Chang reference is rooted in the distinction between the distributed computer system arts and the distribution of media arts. The Appellant submits that the respective teachings are distinct and that the teachings cannot simply be "swapped" between the distributed computer system domains and distribution of media domains. The Examiner, on the other hand, asserts that the teachings of the distributed computer system and distribution of media domains are effectively interchangeable. The issues on appeal are whether the Stefik and Change references teach all the limitations of Appellant's invention, and whether a person of ordinary skill in the art of the invention (portal frameworks) would have found it obvious to substitute the teachings of a distributed computer

system reference for those of a distribution of media reference and to thereby arrive at the invention.

2 Prior Art

The invention of the Application is a method for sharing objects between web sites in a portal framework. In the background of the invention section of the application, the Appellant describes the manner in which objects of a web site were previously implemented. Specifically, in the Background of the Invention the Appellant describes a prior art system in which implementing web objects of a web site provided by a portal requires the physical transfer of files corresponding to web objects to systems which desire to implement these web objects. Alternatively, an entire web site can be archived in a file and transferred to a system in order to implement that web site on the system. Not only are these processes complex, costly and prone to error, but the process requires the manual transfer of code and libraries to a system. (See Application Page 2, Lines 4-15)

Stefik discloses a system for controlling the distribution of digital works, such as movies, music, pictures, multi-media works etc. In the system of Stefik, a "digital ticket" can be used as one element for regulating copying of these digital works. Usage rights associated with usage rights can specify a ticket needed for a particular usage right. (See Stefik Col. 4, Lines 7-25). A transaction involving digital tickets must locate an appropriate digital ticket agent who can validate the ticket before the transaction can proceed. (See Stefik Col. 22, Lines 34-39). A ticket agent can "punch" a digital ticket associated with a digital work. Once a ticket has been punched it cannot be used again for the same kind of transaction unless it is refreshed. (See Stefik Col. 22, Line 60-Col. 23, Line 2) Punching includes marking the ticket with the timestamp of the date and the time it was used. (See Stefik Col. 23, Lines 2-5) The digital ticket can be refreshed when it is copied or extracted, so long as it has copy/extract rights attached. (See Stefik Col. 23, Lines 5-12). When a ticket agent is given a ticket, the agent can check whether the digital copy was made after the last time the ticket was punched. (See *id.*) If a paid copy of a digital work, including a ticket, is made, the new owner would expect to get a fresh ticket. In contrast, loaning a work or simply transferring it to another repository should not revitalize the ticket. (See Stefik Col. 23, Lines 14-27).

Chang discloses a method, system, apparatus and computer program product for a distributed port firewall system. The system of Chang provides a methodology for managing

leases on system resources within a distributed computing environment such as a portion of a computer system's physical units, a portion of the computer system's logical unit's or a portion of the computer system's functionality that is identifiable or addressable in some manner to other physical or logical units within the distributed system. (See Chang Paragraph [0054]) Consumers of these resources can obtain leases on consumable resources such that the resources are made available. (See Chang Paragraph [0055]).

3 Improvement Over Prior Art

The invention of the Application improves upon the prior art by providing a method, system and computer program product which allows an object to be shared by web sites in a portal framework without duplication of the object. In other words, one object may be utilized by multiple web sites in a portal framework by using references to the object. An object may be made available to a web site by storing a reference to the object in a repository associated with the web site. This provides several advantages over the prior art. For instance, multiple web sites can utilize a single object without duplication of the object itself and without transferring any of the code or libraries associated with the object. Further, the system is able to share these objects according to a set of privileges, including privileges associated with the repository, a user, or a site. For example, an administrator with privileges with respect to one site may share an object from this site with another site. In the prior art system objects cannot be shared between sites, therefore it is also the case that objects cannot be shared according to a set of privileges

4 Examiner's / Appellant's Positions Regarding Obviousness

The Examiner asserts that the present invention is obvious from the combination of Stefik's system for the distribution of digital media described above and the distributed computer system of Chang. The Examiner incorrectly asserts that the combination of the Stefik reference and the Chang reference teaches all the limitations of the invention in question, and that a person of ordinary skill in the art of the invention would have combined Stefik's distribution of digital media reference with the distributed computer system reference of Chang to arrive at the invention. The Appellant respectfully submits that not only do Stefik and Chang fail to teach all the limitations of the invention, but additionally, that the teachings in the digital media distribution arts and the distributed computer system arts are distinct and that a person of ordinary skill in the art of portal frameworks would not have combined references from these two

domains as suggested by the Examiner. Additionally, even if one was to combine the inventions of Stefik and Chang the result would be large, unwieldy and unsuitable for its intended purpose.

5 Rejections Under 35 U.S.C. §103

The Examiner rejected all of the claims in the application under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,236,971 ("Stefik") and Publication No. 2002/0078377 ("Chang"). In order to establish a prima facie case of obviousness, the Examiner must show that three criteria are met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. M.P.E.P. 2143. The Appellant respectfully submits that the Examiner has failed to establish all of these criteria. This will be explained below, following a review of the Examiner's arguments.

5.1 Examiner's Reasoning

The Examiner stated in the Office Action that Stefik teaches the invention, except that Stefik does not specifically disclose the object being an invocable software object. More specifically the Examiner has stated that "Chang is used merely to shown [sic] that it is known in the art to provide programs in object forms and to lease computer resources." (See Paper 18, Page 1) The Examiner further states that "given the teaching of Stefik, it would have been obvious of one of ordinary skill in the art to apply the teaching of Stefik to control the leasing of software objects" (See Paper 15, Page 3-4) In other words, the Examiner believes it would be obvious simply to take the system of Stefik and utilize it for controlling the leasing of invocable software objects and other system resources.

The Examiner attempted to show motivation for this change by stating that "it would have been obvious...because it would have enabled the owner of a software object to specify usage and distribution rights to the software object." (See Paper 15, Page 3, Last Paragraph)

5.2 Not All Limitations Disclosed

Independent Claim 1 of the Application recites “storing a reference to an object in a first repository, wherein the object is an invokable software object, performing a first operation to store a duplicate of the reference to the object in a second repository, wherein the first operation is in accordance with a first privilege granted as defined by a permission.” Independent Claim 27 recites “a first repository operable to store a reference to an object wherein the object is an invokable software object, and a second repository operable to store a duplicate of the reference to the object in response to a first operation, wherein the first operation is in accordance with a first privilege granted as defined by a permission.” Independent Claims 14 and 44 recite similar limitations. Thus, each of these independent Claims shares the elements of storing a reference to an object in a first repository and storing a duplicate of the reference to the object in a second repository. Furthermore, each shares the common recitation that the operation to store the duplicate is accordance with a privilege.

The term “reference to an object” as used in the art, is understood to mean a reference that points to the location of an object in a database or file system and identifies the object type, (for example, a pointer). Based on the reference to the object, a programmatic user associated with a site can find the object and invoke an instance of the object. The present invention is drawn to the duplication of the reference to an object from a first repository to a second repository. If a site has access to the second repository, the site will be able to locate the referenced object and invoke an instance of the object based on the duplicated reference to the object, even without access to the first repository. Access to objects can be controlled by requiring that duplication of a reference to an object occur according to a set of permissions. For example, a first site wishing to share an object A, which is referenced in a first repository associated with object A, can only duplicate the reference to a second repository associated with a second site if the first site has privileges to add references to the repository of the second site. If the reference is duplicated to the repository of the second site, the second site can invoke an instance of object A, even without access to the repository of site A. This allows for more efficient administration of multiple sites that require access to common objects.

The Examiner equates the digital ticket of Stefik to the reference to the invokable software object of Claims 1, 14 and 27 of the Application. However, as explained above, the digital ticket of Stefik is used to entitle the holder of the digital ticket to exercise some usage right with respect to a digital work and, hence, acts a portion of a permission scheme. In particular, the

digital ticket of Stefik is a digital certificate such that possession of the certificate is required to gain access to the digital work. (See Stefik Col. 8, Lines 1-5) There is no teaching or suggestion that a particular work can be located in a database from the contents of the digital ticket, that a software object can be invoked using a digital ticket, or that the digital ticket functions as a "reference to an object" as that phrase is used in the art.

More specifically, during a copy operation the requester sends a server a message to initiate a "Copy Transaction". (See Stefik, Col 34, Lines 16-43) The message indicates the work to be copied, the version of the copy right to be used for the transaction, the destination address information for placing the work and the number of copies requested. (See Stefik Col. 34, Lines 23-28) If the work requires a ticket, a ticket may be given and punched by a distributor the work is then copied to the requestor's repository. (See Stefik Col. 46, Lines 45-50). Thus, the desired work must be pointed out with specificity in a message from the requestor, and a ticket submitted separately. These types of tickets are not specific to the digital work requested, and a ticket may be used in transactions with more than one digital work.

An analogy may be made to purchasing tickets at a fair. When one arrives at a fair a number of generic tickets may be purchased. These tickets may be used for a variety of things within the fair, despite the fact that the tickets contain no references to those things. For example, the tickets may be used to ride the Ferris wheel, or the tilt-a-whirl. Additionally, these tickets may be used to purchase funnel cake or cotton candy. Thus, when a ride or food item is desired it must be identified separately, after a ticket can be presented in order to obtain the ride or food item. Thus, though those tickets contain no reference to a particular good or service they are necessary to obtain these goods or services. Additionally, the same ticket may be used to obtain a variety of goods or services.

As the digital ticket of Stefik does not point to the location of an object in a database or file system, does not identify an object type and does cannot be used to invoke an object, but instead is used to obtain usage rights to an already identified work, the digital ticket of Stefik is not a reference to an invokable software object. Accordingly, Applicant respectfully submit that the Examiner has not pointed out where storing a duplicate of a reference to an object as recited in independent Claims 1, 14, 27 and 40 can be found in Stefik.

5.3 Flaws In The Examiner's Reasoning Regarding Combination of Stefik and Chang

The fallacy of the Examiner's reasoning that it would be obvious simply to take the media distribution method of Stefik and apply it to the invokable software objects of Chang is apparent from the fact that there are different distribution and usage patterns for the digital works of Stefik and the invokable software objects of Chang. For example, the digital works of Stefik, including audio, video and electronically published material, may be eventually distributed to human users who may have need to further distribute this digital work to other human users. In contrast, Chang is directed to a methodology for managing leases on system resources where these resources do not need to be further distributed to another recipient, and cannot be duplicated, replicated, or have multiple simultaneous users without defeating the purpose of Chang. Thus, the needs for the distribution system of Stefik differ drastically from the needs for managing system resources of Chang. The system of Stefik must manage the copy and distribution of multiple copies of a duplicable digital work while Chang arranges the serial use of non-duplicable system resources. Because of these factors, a system for distribution of media cannot be translated and applied to a system for the distribution of system resources, nor is it obvious that such a system would even be applicable to the distribution of system resources.

Further, in regard to the Examiner's assertion that the combination of Stefik and Chang "would have been obvious...because it would have enabled the owner of a software object to specify usage and distribution rights to the software object," the Appellant will explain below how applying Stefik to distribute system resources would render the resulting system for the distribution of system resources overly complex and exceedingly expensive due to the high overhead costs imposed.

5.4 Applying the System of Stefik to the Resources of Chang would Result in Needless Complexity

Despite the Examiner's assertion that it would be desirable to apply the invention of Stefik to invokable software objects to enable the owner of an invokable software object to specify usage and distribution rights to a software object, utilizing the invention of Stefik to control access to invokable software objects would create a system with a needlessly complex overhead that would be detrimental to the operation of the system.

More specifically, Stefik refers to usage rights of various digitally encoded works, including audio, video and electronically published material. These digital works are eventually distributed to human users who may have need to further distribute the digital work to other human users. Thus, human users are granted access to these digital work using digital tickets. This may be simultaneous access to a digital work, or copies of the digital work. To control this access, in the usage rights of Stefik are attached directly to digital works (See Stefik Col. 9, Lines 14-16). This may be implemented through the use of a description tree which contains the usage rights of a digital work and makes it possible to examine the rights and fees for a digital work without reference to the contents of the digital work. (See Stefik Col. 9, Lines 29-34) The rights portion will contain a data structure, such a look up table, wherein the various information associated with the a right is maintained. (See Stefik Col. 10, Lines 6-9), and it is "fundamental to the invention that the usage rights are treated as part of the digital work", so the usage rights can be transported with any copy of the digital work. (See Stefik Col. 11, Lines 32-40). These usage rights are defined in a high level "usage rights language" to define usage rights associated with digital works and their parts. (See Stefik Col. 17, Lines 52-55). These usage rights can include the need for a ticket to access the digital work, this ticket must be presented to a digital ticket agent who can punch the digital ticket before access can be gained to a digital work. (See Stefik Col. 22, Line 60-Col. 23, Lines 14)

In contrast, Chang is directed to a methodology for managing leases on system resources within a distributed computing environment, wherein a resource is a portion of a computer system's physical units, logical units, or functionality. These physical and logical resources do not have the usage and distribution patterns of the digital works of Stefik. The resources of Chang exist in a distributed computing environment. (See Chang, Summary of the Invention, Paragraph [0032]) In an environment such as this a software program that has received use of a logical resource has no need to further distribute that resource. These distributed resources are a way of executing logic or functionality on one computer, and after that functionality is executed the distributed local resource may be released, after which the resource may be leased to another software program or user. (See FIG. 9, Paragraph [0087]) Thus, the resources of Chang may be serially assigned to different users, and each of these system resources may be heavily in demand.

To illustrate further, in the system resulting from the combination of Stefik and Chang each software object would need to be associated with a description tree associated with it.

This description tree would contain the usage rights of the invokable software object that make it possible to examine the rights and fees for the invokable software object without reference to the contents of the digital work. The description tree could contain a data structure, such as a look up table, wherein the various information associated with the rights are maintained. This usage tree, in many cases, would be larger than the invokable software object itself, and would have to be passed each time the invokable software object is passed to another. The need to pass this usage tree along with the invokable software object would increase the time it takes to pass the invokable software object, take up more memory, decrease the efficiency and in general place an unneeded burden on the resulting system.

Additionally, every resource within a system which wishes to access the invokable software object would need to obtain a digital ticket, forward this digital ticket to a digital ticket agent before the invokable software object can be accessed or used. Again, the need to pass this digital ticket along with the invokable software object would increase the time it takes to pass the invokable software object, take up more memory, decrease the efficiency and in general place an unneeded burden on the resulting system.

Consequently, the combination of the Stefik and Chang has requirements that result in a needlessly complex system for regulating the use and distribution of an invokable software object which will not be further distributed.

5.5 Examiner Has Failed To Show Suggestion Or Motivation To Combine References

As noted above, in order to make a prima facie case of obviousness, the Examiner must show that there is some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings (see M.P.E.P. 2143). The Examiner implies that it would have been an obvious matter of engineering design choice to apply the invention of Stefik to share invokable software objects (e.g., by stating that "it is known at the time of the invention to share/lease invokable software objects" and "it would have been obvious for one of ordinary skill in the art to apply Stefik's teaching to control distribution of any type of digital work, including...invokable software objects"). While the Examiner cites two passages of Stefik (See Stefik Abstract, Col. 4, Lines 6-14) as motivation to combine the Stefik and Chang references, these passages serve only to describe the functionality of the invention of Stefik, not to suggest that it would be desirable to apply the invention of Stefik to invokable software objects to enable

the owner of a software object to specify usage and distribution rights to a software object. (See *In re Garrett*, 33 BNA PTCJ 43 (November 12, 1986); *In re Chu*, 66 F.3d 292, 36 USPQ2d 1089 (Fed. Cir. 1995))

Contrary to the Examiner's assertion, there is simply no suggestion in the Stefik reference to modify its own structure to be applied to the invokable software objects of Chang. The invention of Stefik is intended to share digital works between users, where these digital works may be copied and simultaneously accessed. In contrast, the invention of Chang is intended to serially lease resources to users in a distributed system (See Chang FIG. 9). The knowledge generally available to one of ordinary skill in the art taught that, in the distributed system domain, a system for distributing system resources should employ as low an overhead as possible protection by avoiding the overhead of systems such as the one presented in Stefik, as elaborated on above in Section 5.4. As there is no direct teaching in Stefik to modify its own structure with the teachings of Chang, and the knowledge of one of ordinary skill in the art would not have taught the combination of the Stefik and Chang references, the suggestion/motivation criterion of M.P.E.P. 2143 is not met.

5.6 Unacceptable Performance of System of Stefik Applied to Chang

The example elaborated on above in Section 5.4 also makes it clear that, in the specific case of the present system for sharing invokable software objects in a portal framework, the Stefik and Chang references could not be combined to form an improved, or additionally even a workable system. Because the overhead associated with distributing an invokable software object would most likely not be acceptable, the system based on the combination of these references would likely not be usable, and would most likely result in a system unable to distribute its resources quickly enough to be usable, and certainly a system which would not be cost effective or efficient. The system would therefore not be considered a successful result of the combination of the references.

5.7 Examiner Has Failed To Show Reasonable Expectation Of Success

In addition to the suggestion/motivation criterion, the Examiner must show that there is a reasonable expectation of success in combining the references (see M.P.E.P. 2143). As explained above, the Examiner's suggested application of the Stefik system to the distributed system domain of Chang would use a description tree (Stefik's implementation) containing the

usage rights for every system resource. As described in Stefik such a description tree would contain a data structure, such as a look up table, wherein the various information associated with the system resource is maintained. As also explained above, the overhead for this type of system would be substantial, and would increase with the number of system resources. Consequently, even if, *arguendo*, a person of ordinary skill were motivated to modify Stefik as suggested by the Examiner, there would be no reasonable expectation of an improved system. In other words, the overhead and complexity of the system suggested by the Examiner would be so great as to make the system ineffective. Thus, the reasonable expectation of success criterion of M.P.E.P. 2143 is not met.

5.8 Examiner Has Failed To Make A Prima Facie Case Of Obviousness Under 35 U.S.C. §103

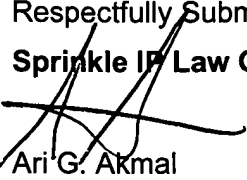
Because the Examiner has failed to show that the prior art references teach or suggest all the claim limitations, that there was a suggestion or motivation to combine the Stefik and Chang references in spite of the contrary teachings, and that there was a reasonable expectation of success in combining the references, the Examiner has failed to meet the criteria set forth in M.P.E.P. 2143 for a prima facie case of obviousness. Accordingly, the Appellant requests that the rejection of the claims under 35 U.S.C. §103 be withdrawn.

6 Conclusion

As explained above, the Appellant believes the Examiner has failed to make a prima facie case of obviousness under 35 U.S.C. §103, and that the corresponding rejection of claims 1-44 should properly be withdrawn. The Appellant therefore respectfully requests that all of the rejections be withdrawn and that all the pending claims be allowed.

Enclosed is a check in the amount of \$500.00 representing the filing fee for the filing of a brief in support of an appeal. While Applicant believes no further fees are due and owing, if Applicant is in error, the Commissioner is hereby authorized to deduct the appropriate amount from Deposit Account No. 50-3183 of Sprinkle IP Law Group.

Respectfully Submitted,
Sprinkle IP Law Group



Ari G. Akmal
Reg. No. 51,388

Date:

1301 W. 25th Street, Suite 408
Austin, TX 78705
Tel. (512) 637-9220
Fax. (512) 317-9088

IX. APPENDIX

1. A method of sharing an object in a portal framework, the method comprising the steps of:

storing a reference to the object in a first repository, wherein the object is an invokable software object; and

performing a first operation to store a duplicate of the reference to the object in a second repository;

wherein the first operation is in accordance with a first privilege granted as defined by a permission.

2. The method according to claim 1, wherein the reference to the object is for an object of a first site.

3. The method according to claim 2, further comprising the step of performing a second operation to add the object to a second site.

4. The method according to claim 3, further comprising the step of performing a third operation to remove the duplicate of the reference of the object from the second repository.

5. The method according to claim 4, wherein the third operation further comprises the step of removing the object from the second site.

6. The method according to claim 2, further comprising the step of providing access to the duplicate of the reference of the object in the second repository.

7. The method according to claim 6, wherein the step of providing access is in accordance with a second privilege granted as defined by a permission for the second repository.

8. The method according to claim 7, further comprising the step of performing a second operation to store a second duplicate of the reference to the object in a third repository

9. The method according to claim 8, further comprising the step of performing a third operation to add the object to a second site.

10. The method according to claim 9, further comprising the step of performing a fourth operation to remove the duplicate of the reference of the object from the second repository.

11. The method according to claim 10, wherein the fourth operation further comprises the step of removing the object from the second site.

12. The method according to claim 2, wherein the first operation further comprises the step of storing references to each child object of the object in the second repository.

13. The method according to claim 2, wherein the first operation further comprises the step of excluding references to at least one child object of the object in the second repository.

14. A computer program product for sharing an object in a portal framework, the computer program product comprising:

a computer readable medium; and

computer program instructions, recorded on the computer readable medium, executable by a processor, for performing the steps of:

storing a reference to the object in a first repository, wherein the object is an invokable software object; and

performing a first operation to store a duplicate of the reference to the object in a second repository;

wherein the first operation is in accordance with a first privilege granted as defined by a permission.

15. The computer program product according to claim 14, wherein the reference to the object is for an object of a first site.

16. The computer program product according to claim 15, further comprising computer program instructions for performing the step of performing a second operation to add the object to a second site.

17. The computer program product according to claim 16, further comprising computer program instructions for performing the step of performing a third operation to remove the duplicate of the reference of the object from the second repository.

18. The computer program product according to claim 17, wherein the third operation further comprises the step of removing the object from the second site.

19. The computer program product according to claim 15, further comprising computer program instructions for performing the step of providing access to the duplicate of the reference of the object in the second repository.

20. The computer program product according to claim 19, wherein the step of providing access is in accordance with a second privilege granted as defined by a permission for the second repository.

21. The computer program product according to claim 20, further comprising computer program instructions for performing the step of performing a second operation to store a second duplicate of the reference to the object in a third repository

22. The computer program product according to claim 21, further comprising computer program instructions for performing the step of performing a third operation to add the object to a second site.

23. The computer program product according to claim 22, further comprising computer program instructions for performing the step of performing a fourth operation to remove the duplicate of reference of the object from the second repository.

24. The computer program product according to claim 23, wherein the fourth operation further comprising the step of removing the object from the second site.

25. The computer program product according to claim 15, wherein the first operation further comprises the step of storing references to each child object of the object in the second repository.

26. The computer program product according to claim 15, wherein the first operation further comprises the step of excluding references to at least one child object of the object in the second repository.

27. A system for sharing an object in a portal framework, the system comprising:
a first repository operable to store a reference to the object, wherein the object is an invokable software object; and
a second repository operable to store a duplicate of the reference to the object in response to a first operation
wherein the first operation is in accordance with a first privilege granted as defined by a permission.

28. The system according to claim 27, further comprising a first site operable to provide an object having to the reference to the object.

29. The system according to claim 28, further comprising a second site operable to provide the object in response to a second operation.

30. The system according to claim 29, wherein the duplicate of the reference of the object is removed from the second repository in response to a third operation.

31. The system according to claim 30, wherein the object is removed from the second site in response to the third operation.

32. The system according to claim 27, wherein access to the duplicate of the reference of the object in the second repository is provided.

33. The system according to claim 32, wherein access is provided in accordance with a second privilege granted as defined by a permission for the second repository.

34. The system according to claim 33, further comprising a third repository operable to store a second duplicate of the reference to the object in response to a second operation.

35. The system according to claim 34, further comprising a second site operable to provide the object in response to the third operation.

36. The system according to claim 35, wherein the second reference of the object is removed from the second repository in response to a fourth operation.

37. The system according to claim 36, wherein the object is removed from the second site in response to the fourth operation.

38. The system according to claim 28, wherein the second repository is operable to store references to each child object of the object in response to the first operation.

39. The system according to claim 28, wherein the second repository is operable to exclude storage of at least one child object of the object in the second repository in response to the first operation.

40. A method of sharing an object in a portal framework comprising:
storing a reference to an object in a first repository, wherein the object is an invokable software object;
storing a first copy of the reference to the object in a second repository;
wherein the first repository and the second repository comprise portions of a single shared repository and wherein the second repository is associated with a first site based on a first set of privileges.

41. The method of Claim 40, further comprising storing a second copy of the reference to the object in a third repository associated with a second site based on a second set of privileges.

42. The method of Claim 42, wherein the third repository comprises another portion of the single shared repository.

43. The method of Claim 40, wherein the object referenced by the reference to the object is not copied to the second repository.

44. The method of Claim 40, further comprising storing the object referenced by the reference to the object in a database remote from the single shared repository.